UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/099,721	03/14/2002	Gregory E. James	NVIDP074/P000427	1906

28875   7590   07/12/2007
Zilka-Kotab, PC
P.O. BOX 721120
SAN JOSE, CA 95172-1120

EXAMINER
GUILL, RUSSELL L

ART UNIT	PAPER NUMBER
2123	

MAIL DATE	DELIVERY MODE
07/12/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/099,721	JAMES, GREGORY E.
	Examiner	Art Unit	
	Russ Guill	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 May 2007*.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4,5,7-15,17,18 and 20-31* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4,5,7-15,17,18 and 20-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *14 March 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date *5/23/2007*.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____ .

U.S. Patent and Trademark Office

PTOL-326 (Rev. 08-06)   Office Action Summary   Part of Paper No./Mail Date 20070704

DETAILED ACTION

1.   This non-final Office Action is in response to an <u>Amendment</u> filed May 29, 2007. Claims 16 and 19 were canceled. No claims were added. Claims 1 - 2, 4 - 5, 7 - 15, 17 - 18 and 20 – 31 are pending. Claims 1 - 2, 4 - 5, 7 - 15, 17 - 18 and 20 – 31 have been examined. Claims 1 - 2, 4 - 5, 7 - 15, 17 - 18 and 20 – 31 have been rejected.

2.   **This Office Action is NON-final due to new rejections.**

Response to Remarks

3.   Regarding claims 1-2, 4-5, and 7-31 rejected under 35 USC 101:

   a.   Applicant's arguments have been fully considered, and are persuasive.

4.   Regarding claims 1-2, 4-5, and 7-31:

   a.   Applicant's arguments have been fully considered, and are persuasive for claims 18 and 22, as follows; however, new rejections are made below. The remaining arguments are not persuasive, as follows.

   b.   The Applicant argues:

   c.   With respect to Claims 1-2, 4-5, and 7-31, the Examiner has rejected the same under 35 U.S.C. 103(a) as being unpatentable over Rumpf ("Using Graphics Cards for Quantized FEM Computations"). First, the Examiner has argued that "the art of Rumpf teaches using a graphics hardware pipeline to solve partial differential equations" and that therefore "implementing any known method of solving a partial differential equation using a hardware graphics pipeline would have been obvious."

   d.   Applicant respectfully disagrees, and asserts that Rumpf merely teaches "a graphics hardware solver for the linear heat equation" (Section 6, first column, paragraph 1) with "a defined linear system of equations... [that] have [been] shown to be able to [be] solve[d]... in graphics hardware" (Section 6, second column, paragraph 4). Furthermore, Rumpf teaches that "we had to <u>approximate all involved nonlinear functions by linear in the implementation</u> of the anisotropic diffusion" which "<u>leads to an deterioration in image quality</u> in the following timesteps," and that "the <u>restricted precision of bits per color component</u> leads to <u>unsatisfying results for the linear heat equation</u> because smooth transitions in temperature produce very small values in the convolution, with <u>very high relative errors</u>" (Section 8, second column, paragraph 3 -- emphasis added).

   e.   Clearly, the disclosure of having to <u>approximate all nonlinear functions with linear functions</u> which lead to <u>a deterioration in image quality</u>, in addition to a restricted precision of bits per color component which lead to <u>unsatisfying results</u> and

<u>very high relative errors</u> for the linear heat equation, as in Rumpf, simply fails
support the Examiner's argument that it would be obvious to implement "any known method
of solving a partial differential equation using a hardware graphics pipeline." Applicant
thus formally requests a specific showing of the subject matter in ALL of the claims in
any future action. Note excerpt from MPEP below.

     i.   "If the applicant traverses such an [Official Notice] assertion the
         examiner should cite a reference in support of his or her position." See MPEP
         2144.03.

     ii.   The Examiner respectfully replies:

     iii.   The Examiner acknowledges that Rumpf discloses the "important issues" recited
by the Applicant. However, Rumpf continues to disclose, "But the remaining restrictions
do not distract us from the <u>looming possibilities</u>" (section 8, second column). Rumpf
further teaches, "The presented strategy opens up a wide area of numerical applications
for hardware acceleration. The implementations of Finite Element solvers for the linear
heat equation and the anisotropic diffusion method in image processing underline <u>its</u>
<u>practicability</u>" (Abstract, first paragraph). It is obvious that Rumpf is disclosing
"important issues" rather than teaching away from the invention.

     iv.   Further, the new reference by Wang ("A Processor Architecture for 3D
Graphics", September 1992, IEEE Computer Graphics & Applications) discloses a
graphics pipeline with a suggestion that it can be used to solve partial differential
equations (*page 97, first column*).

     v.   Therefore, the Examiner maintains that in view of Rumpf, it would have been
obvious to implement any known method of solving a partial differential equation using
a hardware graphics pipeline.

f.   The Applicant argues:

g.   Second, the Examiner has argued that "since a graphics pipeline performs numeric
calculation, it is inherent in the device that it can be used to solve a partial
differential equation." Applicant respectfully disagrees and again asserts that Rumpf's
disclosure of "a deterioration in image quality," in addition to "unsatisfying results
and very high relative errors," fails to support the Examiner's inherency argument. Thus,
in view of the arguments made hereinabove, any such inherency argument has been
adequately rebutted, and a notice of allowance or a specific prior art showing of such
claim features, in combination with the remaining claim elements is respectfully
requested. (See MPEP 2112)

    i.   The Examiner respectfully replies:

    ii.   First, the Examiner refers to the previous response. Second, the graphics pipeline

in Rumpf was actually used to solve a partial differential equation, which clearly

supports the assertion that it is inherent in a numeric calculation device that it can be

used to solve a partial differential equation.

h. The Applicant argues:

i. Further, in response, applicant asserts that the fact that a certain result or characteristic
may occur or be present in the prior art is not sufficient to establish the inherency of that
result or characteristic. *in re Rijckaert,* 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir.
1993); *In re Oelrich,* 666 F.2d 578, 581-82, 212 USPQ 323, 326 (CCPA 1981). "To establish
inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is
necessarily present in the thing described in the reference, and that it would be so recognized
by persons of ordinary skill. Inherency, however, may not be established by probabilities or
possibilities. The mere fact that a certain thing may result from a given set of
circumstances is not sufficient" *in re Robertson,* 169 F.3d 743, 745, 49 USPQ2d 1949,
1950-51 (Fed. Cir. 1999).

    i.   The Examiner respectfully replies:

    ii.   The Examiner agrees with the preceding argument. However, in the specific case

under consideration, inherency is clearly established.

5.   Regarding claims 1-2, 10-18, 22-23, and 27 rejected under 35 USC § 103:

    a. The Applicant argues:

b. To establish *a prima facie* case of obviousness, three basic criteria must be met.
First, there must be some suggestion or motivation, either in the references themselves
or in the knowledge generally available to one of ordinary skill in the art, to modify
the reference or to combine reference teachings. Second, there must be a reasonable
expectation of success. Finally, the prior art reference (or references when combined)
must teach or suggest all the claim limitations. The teaching or suggestion to make the
claimed combination and the reasonable expectation of success must both be found in the
prior art and not based on applicant's disclosure. *In re Vaeck,947* F.2d 488, 20 USFQ2d
1438 (Fed.Cir.1991).

C. With respect to the first element of the *prima facie* case of obviousness, the Examiner
has stated that "the motivation to use the art of Rumpf with the art of Press would have
been the benefits recited in Rumpf that the presented strategy opens a wide area of
numerical applications for hardware acceleration *(first page, Abstract, first paragraph)*,
and turns a graphics card into an ultrafast vector coprocessor *(first page, Abstract.
first paragraph)*, which would have been recognized by the ordinary artisan as benefits

that allow faster processing." Applicant respectfully disagrees with this proposition,
especially in view of the vast evidence to the contrary.

      i.   The Examiner respectfully replies:

      ii.   The Applicant may, of course, disagree with the proposition; however, the
benefits recited are from the reference, and therefore, provide valid motivations.

d.   The Applicant argues:

e.   For example, Press relates to <u>implementing mathematics in software,</u> while
Rumpf relates to <u>using graphics cards for quantized FEM, computations</u>. To simply glean
features from a system for performing <u>quantized FEM computations using graphics cards</u>,
such as that of Rumpf, and combine the same with the *non-analogous* art of <u>software-</u>
<u>implemented mathematics</u>, such as that of Press, would simply be improper. <u>Graphics cards</u>
provide broad access to <u>graphics memory</u> and parallel processing of <u>image operands</u>
(see the Abstract of Rumpf), while <u>software-implemented mathematics</u> merely relates to
using software to carry out mathematical operations. "In order to rely on a reference as
a basis for rejection of an applicant's invention, the reference must either be in the
field of applicant's endeavor or, if not, then be reasonably pertinent to the particular
problem with which the inventor was concerned." In re Oetiker, 977 F.2d 1443, 1446, 24
USPQ2d 1443, 1445 (Fed. Cir. 1992). See also In re Deminski, 796 F.2d 436, 230 USPQ 313
(Fed. Cir. 1986); In re Clay, 966 F.2d 656, 659, 23 USPQ2d 1058, 1060-61. (Fed. Cir.
1992). In view of the vastly different types of problems software-implemented mathematics
addresses as opposed to <u>graphics cards</u>, the Examiner's proposed combination is clearly
inappropriate.

      i.   The Examiner respectfully replies:

      ii.   The art of Rumpf and the art of Press are clearly analogous art for at least the
reason that they both pertain to the art of solving partial differential equations (*Press,*
page 838, section 19.2, Diffusive Initial Value Problems; and Rumpf, section 6, Linear Heat
Equation).

f.   The Applicant argues:

g.   In addition, applicant respectfully asserts that the software mathematics of the Press
and Press2 references are implemented using "Fortran 77" and "C" (see respective Titles),
but are not disclosed to be directly on a graphics card. For example, Page 860 of Press
discloses implementing "a routine for SOR with Chebyshev acceleration" in
Fortran. Further, Rumpf discloses having to "approximate all involved nonlinear
functions by linear in the implementation of the anisotropic diffusion" which "<u>leads to</u>
<u>an deterioration in image quality</u> in the following timesteps," and that "the
<u>restricted precision of bits per color component</u> leads to <u>unsatisfying results for the</u>

linear heat equation... with <u>very high relative errors</u>" (Section 8, second column, paragraph 3 - emphasis added). Again, applicant respectfully asserts that the Examiner's proposed combination is clearly inappropriate in view of the vastly different types of problems addressed by <u>software-implemented mathematics</u> as opposed to those addressed by <u>graphics card-implemented quantized FEM computations</u>.

    i.   The Examiner respectfully replies:

    ii.   Again, the Examiner asserts that the art of Rumpf and the art of Press are clearly analogous art for at least the reason that they both pertain to the art of solving partial differential equations (*Press, page 838, section 19.2, Diffusive Initial Value Problems; and Rumpf, section 6, Linear Heat Equation*). Further, hardware and software are equivalent (please refer to the new reference by Tanenbaum, Structured Computer Organization, 1984, page 11).

h. The Applicant argues:

i. Furthermore, Rumpf discloses that "many numerical algorithms <u>still disregard hardware issues</u> and <u>little humps in the graphics hardware</u> **still obstruct the passage to general fast numerical computations**" (Section 1, Goals, paragraph 4 - emphasis added). Applicant asserts that Rumpf's disclosure that many algorithms disregard hardware issues and that graphics hardware <u>obstructs passage</u> to general fast numerical computations clearly *teaches away* from the software-implemented <u>general mathematics</u> of the Press references. *in re Hedges, 783 Eli 1038, 228 USPQ 685 (Fed. Cir. 1986).*

    i.   The Examiner respectfully replies:

    ii.   Following Rumpf's disclosure recited above, Rumpf continues, "Hence even minor considerations of graphics hardware issues with respect to numerics on one side, and development of slightly more hardware sensitive algorithms on the other, could **result in revolutionary speedups for many applications.**" Hence, Rumpf is clearly advocating the method for improving performance of applications.

j. The Applicant argues:

k. Applicant respectfully asserts that at least the first element of the *prima facie* case of obviousness has not been met, since it would be *unobvious* to combine the references, as noted above.

    i.   The Examiner respectfully replies:

ii.   As discussed above, Rumpf and Press are analogous art, and Rumpf provides

motivation to use the art of Rumpf with the art of Press. Therefore, it appears that it

would have been obvious to combine the art of Rumpf and the art of Press.

l.   The Applicant argues:

m.   Nevertheless, despite such paramount deficiencies and in the spirit of expediting the
prosecution of the present application, applicant has at least substantially incorporated
the subject matter of former Claims 16, 19, and 21 into at least some of the independent
claims.

n.   With respect to the subject matter of former Claim 16 (now at least substantially
incorporated into at least some of the independent claims), the Examiner has relied on
Page 855 from the Press reference to make a prior art showing of applicant's claimed
technique "wherein the processing further includes determining whether the solution has
converged" (see this or similar, but not necessarily identical language in at least some
of the independent claims).

O.   Applicant respectfully disagrees and asserts that the excerpt from Press relied
upon by the Examiner merely discloses that "the algorithm consists of using the average
of u at its four nearest-neighbor points on the grid" and that "[t]his procedure is then
iterated until convergence." However, averaging u at it's four nearest-neighbor
points until convergence, as in Press, simply fails to suggest a technique
"wherein the processing further includes determining whether the solution has converged"
(emphasis added), as claimed by applicant. Clearly, repeating the averaging until
convergence, as in Press, simply fails to even suggest that "processing further includes
determining whether the solution has converged" (emphasis added), where the processing
"utiliz[es] the hardware graphics pipeline," in the context as claimed by applicant (see
the same or similar, but not necessarily identical language in the independent claims,
for context).

i.   The Examiner respectfully replies:

ii.   As an initial remark, the recited method from Press is Jacobi's method (page 855,

the paragraph that starts with, "This method is in fact a classical method . . ."), which is

also used by Rumpf (section 5, "Typical solvers are the the Jacobi iteration").

iii.   The Applicant merely asserts that averaging u at it's four nearest-neighbor points

until convergence, as in Press, simply fails to suggest a technique "wherein the

processing further includes determining whether the solution has converged", but does

not specifically point out how the language of the claims distinguishes from the

reference. Typically, solving partial differential equations with a computer involves

discretizing the equations, and then solving a set of linear equations (see Rumpf, sections

3.1 and 5). As in Rumpf, section 5, linear equations are often solved using an iterative

method. The Jacobi iterative method of Rumpf, section 5, is repeated until convergence, as described in Press, page 855. Therefore, the limitation appears to be satisfied by Press, as described in the Office Action.

p. The Applicant argues:

q. Further, with respect to the subject matter of former Claim 19 (now at least substantially incorporated into at least some of the independent claims), the Examiner has relied on Page 400 from the Roy-Chowdhury reference to make a prior art showing of applicant's claimed technique "wherein the determining whether the solution has converged includes calculating errors" (see this or similar, but not necessarily identical language in at least some of the independent claims).

r. Applicant respectfully asserts that the excerpt from Roy-Chowdhury relied upon by the Examiner merely discloses that "[t]he expressions for updating errSR_and errSB_ in each iteration... may be derived by summing over all red and black points." Further, Roy-Chowdhury discloses that "wherever error bounds for individual elements of u[i][j] arise in our error expressions, we drop them." However, the mere disclosure of updating errSR_ and errSB_ in each iteration, and dropping error bounds for individual elements when they arise in the error expressions, as in Roy-Chowdhury, simply fails to suggest a technique "wherein the determining whether the solution has converged includes calculating errors" (emphasis added), as claimed by applicant.

i.   The Examiner respectfully replies:

ii.   The excerpt from Roy-Chowdhury needs to be evaluated in the context of the reference, especially the teaching that the termination condition for the iterative method is determined at runtime by specifying that the outer loop continue until the maximum difference over all grid points of a point value at the current iteration from its value at a previous iteration drops below a threshold (*page 395, right-side column, top half*). In combination with the context, the disclosure for updating errSR_ and errSB_ in each iteration suggests determining whether the solution has converged includes calculating errors. Further, the specification appears to be silent on the meaning of "calculating errors."

s. The Applicant argues:

t. Additionally, with respect to the subject matter of Claim 21 (now at least substantially incorporated into at least some of the independent claims), the Examiner has relied on Page 400 from the Roy-Chowdhury reference to make a prior art showing of applicant's claimed technique "wherein the determining whether the solution has converged further includes concluding that the solution has converged if the error is less than a

predetermined amount." Applicant respectfully points out that, as currently amended, at
least some of the independent claims now claim a technique "wherein the determining
whether the solution has converged further includes concluding that the solution has
converged based on the calculation of the errors" (see this or similar, but not
necessarily identical language in at least some of the independent claims).

u.   Applicant respectfully asserts that the excerpt relied on by the Examiner fails to
meet applicant's claim language, as amended. For example, the excerpt from RoyChowdhury
relied upon by the Examiner merely discloses that "the expressions for updating errSR and
errSB_ in each iteration... may be derived by summing over all red and black points."
Further, Roy-Chowdhury discloses that "wherever error bounds for individual elements of
u[i][j] arise in our error expressions, we drop them." However, the mere disclosure of
updating errSR_ and errSB_ in each iteration, and dropping error bounds for individual
elements when they arise in the error expressions, as in RoyChowdhury, simply fails to
suggest a technique "wherein the determining whether the solution has converged further
includes concluding that the solution, has converged based on the calculation of said
errors" (emphasis added), as claimed by applicant.

v.   Furthermore, applicant notes that Row-Chowdhury discloses that "[I]n this paper, we develop
low-overhead, error-detecting versions of iterative algorithms for solving the regular, spare
linear systems which arise from discretizations of various partial differential
equations (PDEs)" (Page 394, second column – emphasis added). Clearly, disclosing error-
detecting versions of iterative algorithms, as in Row-Chowdhury, simply fails to even suggest.
"concluding that the solution has converged based on the calculation of said errors" (emphasis
added), in the manner as claimed by applicant.

i.   The Examiner respectfully replies:

ii.   The excerpt from Roy-Chowdhury needs to be evaluated in the context of the
reference, especially the teaching that the termination condition for the iterative method
is determined at runtime by specifying that the outer loop continue until the maximum
difference over all grid points of a point value at the current iteration from its value at a
previous iteration drops below a threshold (*page 395, right-side column, top half*). In
combination with the context, the disclosure for updating errSR_ and errSB_ in each
iteration suggests determining whether the solution has converged includes concluding
that the solution has converged based on the calculation of said errors. Further, the
specification appears to be silent on the meaning of "calculating errors."

w. The Applicant argues:

x.   Still yet, with respect to independent Claims 28 and 29, applicant respectfully asserts that
such claims are deemed novel in view of the prior art excerpts relied on by the Examiner for at
least substantially the same reasons argued above. For example, Claim 28 recites "determining
whether the solution has converged," as claimed, which is clearly not met by the prior art

reference excerpts relied on by the Examiner, as noted above, in addition, Claim 29 recites "determining whether the solution has converged by: calculating errors, summing the errors, and concluding that the solution has converged if the sum of errors is less than a predetermined amount," as claimed, which is also clearly not met by the prior art reference excerpts relied on by the Examiner, for substantially the same reasons as noted above.

    i.   The Examiner respectfully replies:

    ii.   Since claims 28 and 29 are argued for substantially the same reasons argued above, please refer to the above Examiner's replies.

y. The Applicant argues:

z.   To this end, applicant respectfully asserts that at least the first and third elements of the *prima facie* case of obviousness have not been met, since it would be *unobvious* to combine the references, as noted above, and the prior art reference excerpts, as relied upon by the Examiner, fail to teach or suggest <u>all</u> of the claim limitations, as noted above. Thus, a notice of allowance or a proper prior art showing of <u>all</u> of applicant's claim limitations, in combination with the remaining claim elements, is respectfully requested.

    i.   The Examiner replies:

    ii.   As discussed above, it would have been obvious to combine the references, and the prior art appears to teach or suggest all of the claim limitations.

aa. The Applicant argues:

bb. Applicant further notes that the prior art is also deficient with respect to the dependent claims. For example, with respect to Claim 18, the Examiner has relied on Page 855 from the Press reference to make a prior art showing of applicant's claimed technique "wherein it is determined whether the solution has converged after a predetermined number of multiple iterations of the relaxation operation."

cc. Applicant respectfully asserts that the excerpt from Press relied upon by the Examiner merely discloses that "the algorithm consists of using the average of u at its four nearest-neighbor points on the grid" and that "[t]his procedure is then iterated until convergence." Further, Press discloses that "[t]is method is in fact a classical method... called *Jacobi 's method*" However, averaging u at it's four nearest-neighbor points until convergence, as in Press, simply fails to suggest a technique "wherein it is determined whether the solution has converged <u>after a **predetermined number** of multiple iterations</u> of the relaxation operation" (emphasis added), as claimed by applicant. Clearly, the mere disclosure of <u>iterating until convergence</u>, as in Press, simply fails to even suggest "a <u>predetermined number</u> of multiple iterations" (emphasis added), in the manner as claimed by applicant.

     i.   The Examiner respectfully replies:

     ii.   Applicant's argument has been fully considered, and is persuasive. However, a new rejection is made below.

dd. The Applicant argues:

ee. Further, with respect to Claim 22, the Examiner has relied on Pages 862-868 in Press to make a prior art showing of applicant's claimed technique "wherein if it is determined that the solution has converged, repeating the processing using an altered parameter value." Specifically, the Examiner has argued that "it would have been obvious [that] altering a grid size is altering a parameter."

ff. Applicant respectfully disagrees. Press explicitly discloses the "multigrid as an iterative scheme, where one starts with some initial guess on the finest grid and carries out enough cycles . . . to achieve convergence," and that the "simplest way to use multigrid... [is] to apply enough cycles until some appropriate convergence criterion is met" (see Page 868-Full Multigrid Algorithm-emphasis added). In fact, applicant notes that Press even teaches that "[o]ne iteration of a multigrid method, from finest grid to coarser grids and back to finest grid again, is called a cycle" (see Page 865).

gg. Thus, Press clearly discloses carrying out multiple cycles, where each cycle utilizes a plurality of different grid sizes, in order to achieve convergence, which simply does not support the Examiner's argument that: "it would have been obvious [that] altering a grid size is altering a parameter," especially in view of applicant's claimed technique, namely "wherein if it is determined that the solution has converged, repeating the processing using an altered parameter value" (emphasis added), as claimed. Again, applicant formally requests a specific showing of the subject matter in ALL of the claims in any future action. See MPEP 2144.03.

     i.   The Examiner replies:

     ii.   Applicant's argument has been fully considered, and is persuasive. However, a new rejection is made below. The specification appears to provide "a time value" as an example of a parameter (*page 5, line 9*), and this is taken into consideration in the new rejection.

Claim Rejections - 35 USC § 103

6.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections

set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102
> of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject
> matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the
> art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was
> made.

7.   Regarding all **claims 1 - 2, 4 - 5, 7 - 15, 17 - 18 and 20 – 31**, the art of Rumpf (Martin Rumpf et al.;

"Using Graphics Cards for Quantized FEM Computations") teaches using a graphics hardware pipeline

to solve partial differential equations. After the inventive step of Rumpf, implementing any known

method of solving a partial differential equation using a hardware graphics pipeline would have been

obvious. The ordinary artisan would have known to turn to references describing solution methods to

partial differential equations both by the nature of the problem and the benefit of saving time and cost by

using proven previous solution methods. Further, the reference by Wang ("A Processor Architecture for

3D Graphics", September 1992, IEEE Computer Graphics & Applications) discloses a graphics pipeline

with a suggestion that it can be used to solve partial differential equations (*page 97, first column*). Further,

since a graphics pipeline performs numeric calculation, it is inherent in the device that it can be used to

solve a partial differential equation (see MPEP section 2112).

8.   Claims **1 – 2, 12 – 15, 17 - 18, 20 – 23 and 27** are rejected under 35 U.S.C. 103(a) as being unpatentable

over Press (Press, William H.; Flannery, Brian P.; Teukolsky, Saul A.; Vetterling, William T.; "Numerical

Recipes in Fortran 77", 2001, Second edition, Cambridge University Press) in view of Rumpf (Martin

Rumpf et al.; "Using Graphics Cards for Quantized FEM Computations", September 3 – 5 2001,

Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing),

further in view of Roy-Chowdhury (Roy-Chowdhury, Amber; Bellas, Nikolas; Banerjee, Prithviraj;

"Algorithm-Based Error-Detection Schemes for Iterative Solution of Partial Differential Equations", 1996,

IEEE Transactions on Computers, Vol. 45, No. 4).

    a.   The art of Press is directed to numerical methods for solving partial differential equations
    (*page 838, section 19.2*).
    b.   The art of Rumpf is directed to the art of solving partial differential equations using a
    graphics card (*Title and Abstract*).
    c.   The art of Roy-Chowdhury is directed to algorithm based error detection schemes for
    iterative solution of partial differential equations (*page 394, Title*).

d. The art of Rumpf and the art of Press are analogous art at least because they are both pertain to the art of solving partial differential equations.

e. The art of Rumpf and the art of Roy-Chowdhury are analogous art at least because they both pertain to the art of solving partial differential equations.

f. Regarding **claim 1**:

g. Press appears to teach:

>   i.   Receiving input (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that input is required to solve a partial differential equation, especially given the statement that an initial distribution relaxes to an equilibrium distribution on page 855**);

>   ii.   Processing the input to generate the solution to the partial differential equation (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems**);

>   iii.   Wherein the processing further includes determining whether the solution has converged (**pages 855, Relaxation Methods for Boundary Value Problems; second paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."**);

h. Press does not specifically teach:

>   i.   Receiving input *in the hardware graphics pipeline*;

>   ii.   Processing the input to generate the solution to the partial differential equation *utilizing the hardware graphics pipeline*;

>   iii.   *Generating output utilizing the hardware graphics pipeline for display*;

>   iv.   *Wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline*.

>   v.   *Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline*.

>   vi.   *Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures*;

vii. _Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors._

i. Rumpf appears to teach:

i. Receiving input in the hardware graphics pipeline (_third page, figure 1_);

ii. Processing the input to generate the solution to the partial differential equation utilizing the hardware graphics pipeline (_third page, section 3.1 Vector Representation, first paragraph; and seventh page, section 6. Linear Heat Equation, first paragraph_);

iii. Generating output utilizing the hardware graphics pipeline for display (_ninth page, figure 3_);

iv. Wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (_seventh page, section 6. Linear Heat Equation, first paragraph_);

v. Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline (_ninth page, figure 3, displays surfaces and objects rendered by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline_).

vi. Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures (_second and third pages, section 2. Computational Setting; and third page, figure 1_).

j. Roy-Chowdhury appears to teach:

i. Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors (_page 395, right-side column, sentence that starts with, "The termination condition is determined . . .", and page 400, left-side column, top-half; please note that Press also appears to teach this limitation on page 855 second paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."_).

k. The motivation to use the art of Rumpf with the art of Press would have been the benefits recited in Rumpf that the presented strategy opens a wide area of numerical applications for hardware acceleration (*first page, Abstract, first paragraph*), and turns a graphics card into an ultrafast vector coprocessor (*first page, Abstract, first paragraph*), which would have been recognized by the ordinary artisan as benefits that allow faster processing.

l. The motivation to use the art of Roy-Chowdhury with the art of Press would have been the benefit recited in Roy-Chowdhury that the presented algorithm-based fault tolerance is an inexpensive method of achieving fault tolerance without requiring any hardware modifications, especially for iterative solution of linear systems arising from discretization of partial differential equations (**page 394, Abstract**), which would have been recognized as a benefit by the ordinary artisan.

m. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Rumpf and the art of Roy-Chowdhury with the art of Press to produce the claimed invention.

==

n. Regarding claim 2:

o. Press appears to teach:

> i.   Input represents boundary conditions (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that boundary conditions are required to solve a partial differential equation, especially since the title of the section recites Boundary Value problems**);

==

p. Regarding claim 12:

q. Press appears to teach:

> i.   The processing includes a relaxation operation (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that**

processing includes a relaxation operation, especially since the title of the section recites Relaxation Methods);

===

r. Regarding **claim 13**:

s. Press appears to teach:

    i.   The relaxation operation is selected based on the partial differential equation (pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that the relaxation operation is selected based on the partial differential equation, especially since such an example is presented);

===

t. Regarding **claim 14**:

u. Press appears to teach:

    i.   The processing includes a plurality of iterations of the relaxation operation (pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; especially references to Gauss-Seidel method and Jacobi's method);

===

v. Regarding **claim 15**:

w. Press appears to teach:

    i.   A number of iterations of the relaxation operation is reduced using at least one of a prolongation operation and a restriction operation (pages 862-868, section 19.6 Multigrid Methods for Boundary Value Problems, especially page 865 Smoothing, Restriction and Prolongation Operators);

===

x. Regarding **claim 17**:

y. Press appears to teach:

> i.   It is determined whether the solution has converged after each iteration of the
> relaxation operation (<u>pages 855, Relaxation Methods for Boundary Value Problems;</u>
> <u>second paragraph, section that starts with "Thus the algorithm consists . . .", sentence,</u>
> <u>"This procedure is then iterated until convergence."</u>);

==

z.   Regarding **claim 18**:

aa. Press appears to teach:

> i.   It is determined whether the solution has converged after iterations of the
> relaxation operation (<u>page 855, Relaxation Methods for Boundary Value Problems;</u>
> <u>second paragraph, section that starts with "Thus the algorithm consists . . .", sentence,</u>
> <u>"This procedure is then iterated until convergence."</u>);

bb. Press does not specifically teach:

> i.   It is determined whether the solution has converged *after a predetermined*
> *number of multiple* iterations of the relaxation operation.

cc. Official Notice is taken that determining whether a solution has converged after a
predetermined number of multiple iterations was old and well known at the time of invention in
the analogous art of numerical methods. At the time of invention it would have been obvious to
a person of ordinary skill in the art to determine whether the solution has converged *after a*
predetermined number of multiple iterations of the relaxation operation. The motivation would
have been the knowledge of the ordinary artisan that processing time would be saved by testing
convergence only after multiple iterations for a process that takes multiple iterations to converge.
As support for the Official Notice, three references are provided:

> i.   E. Galligani et al.; "Implementation of Splitting Methods for Solving Block
> Tridiagonal Linear Systems on Transputers", 1995, Proceedings of Euromicro Workshop
> on Parallel and Distributed Processing, pages 409 – 415, <u>especially page 412, left-side</u>
> <u>column, sentence that starts with, "The overheads can be minimized . . ."</u>
>
> ii.   Olav Beckmann et al.; "Data Distribution at Run-Time: Re-Using Execution
> Plans", 1998, Euro-Par'98, LNCS 1470, Springer-Verlag, pages 413 – 421, <u>especially page</u>
> <u>418, text for Table 1, convergence test every 10 iterations.</u>

       iii.   Y. Saad; "Krylov Subspace Methods for Solving Large Unsymmetric Linear

Systems", July 1981, Mathematics of Computation, Volume 37, Number 155, pages 105 -

126; teaches testing for convergence every q steps, page 113.

dd. Therefore, it would have been obvious to modify Press and Rumpf to obtain the invention as

specified in claim 18.

==

ee. Regarding **claim 20**:

ff. Press does not specifically teach:

       i.   The determining whether the solution has converged further includes summing

the errors;

gg. Roy-Chowdhury appears to teach:

       i.   The determining whether the solution has converged further includes summing

the errors (**page 395, right-side column, sentence that starts with, "The termination

condition is determined . . .", and page 400, left-side column, top-half**);

==

hh. Regarding **claim 21**:

ii. Press does not specifically teach:

       i.   Concluding that the solution has converged if an error is less than a

predetermined amount;

jj. Roy-Chowdhury appears to teach:

       i.   Concluding that the solution has converged if an error is less than a

predetermined amount (**page 395, right-side column, sentence that starts with, "The

termination condition is determined . . .", and page 400, left-side column, top-half**);

==

kk. Regarding **claim 22**:

ll. Press appears to teach:

> i. If it is determined that the solution has converged repeating the processing using an altered parameter value operation (**pages 838-840, section 19.2 Diffusive Initial Value Problems; especially note on page 840 below equation 19.2.12, the reference to stepsize Δt. The specification appears to provide a time value as an example of a parameter on page 5, line 9**);

===

mm. Regarding **claim 23**:

nn. Press appears to teach:

> i. The number of iterations of the relaxation operation is determined prior to the processing (**pages 860, Relaxation Methods for Boundary Value Problems; code example with a parameter value MAXITS = 1000 and a loop DO N=1,MAXITS**);

===

oo. Regarding **claim 27**:

> i. Claim 27 is taught as in claim 1 above.

===

9.   Claims **4 – 5** are rejected under 35 U.S.C. 103(a) as being unpatentable over Press as modified by Rumpf and Roy-Chowdhury as applied to claims **1 – 2, 12 – 15, 17 - 18, 20 – 23 and 27** above, further in view of Weiskopf (Weiskopf, Daniel; Hopf, Matthias; Ertl, Thomas; "Hardware-Accelerated Visualization of Time-Varying 2D and 3D Vector Fields by Texture Advection via Programmable Per-Pixel Operations", 2001, Proceedings of the Vision Modeling and Visualization Conference 2001).

a.   Press as modified by Rumpf and Roy-Chowdhury teaches a hardware graphics pipeline implemented method for generating a solution to a partial differential equation in a hardware graphics pipeline.

b.   The art of Weiskopf is directed to hardware accelerated visualization of time-varying 2D and

3D vector fields by texture advection (_Title_).

c.   The art of Weiskopf and the art of Press as modified by Rumpf and Roy-Chowdhury are

analogous art because they both contain the art of performing calculations using a graphics card

(_Rumpf, Abstract; Weiskopf, pages 668 – 670, section 3.1 Basic Advection_).

d.   The motivation to use the art of Weiskopf with the art of Press as modified by Rumpf and

Roy-Chowdhury would have been the benefits recited in Weiskopf including the advantage that

the algorithm has extremely high simulation and rendering speed (_page 672, right-side column,

fourth paragraph that starts with, "An advantage . . ."_).

e.   Regarding **claim 4**:

f.   Press does not specifically teach:

   i.   the input includes geometry;

g.   Weiskopf appears to teach:

   i.   the input includes geometry (**pages 668 - 669, section 3 Hardware-Based 2D**

   Texture Advection; it would have been obvious that the input includes geometry;

   please note that the partial differential equation on page 667, right-side column,

   second paragraph, is being solved);

h.   Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time

of invention to use the art of Weiskopf with the art of Press as modified by Rumpf and Roy-

Chowdhury to produce the claimed invention.

==

i.   Regarding **claim 5**:

j.   Press does not specifically teach:

   i.   the geometry is selected from the group consisting of polygons, vertex data,

   points, and lines;

k.   Weiskopf appears to teach:

   i.   the geometry includes points (**pages 668 - 669, section 3 Hardware-Based 2D**

   Texture Advection; it would have been obvious that the input includes geometry;

   please note that the partial differential equation on page 667, right-side column,

   second paragraph, is being solved);

==

10. **Claims 7 – 9 and 24 - 25** are rejected under 35 U.S.C. 103(a) as being unpatentable over Press as

modified by Rumpf and Roy-Chowdhury as applied to claims **1 – 2, 12 – 15, 17 - 18, 20 – 23 and 27** above,

further in view of Ewins (Ewins, Jon P.; Waller, Marcus D.; White, Martin; Lister, Paul F.; "MIP-Map

Level Selection for Texture Mapping", 1998, IEEE Transactions on Visualization and Computer Graphics,

Vol. 4, No. 4).

    a.   Press as modified by Rumpf and Roy-Chowdhury teaches a hardware graphics pipeline

implemented method for generating a solution to a partial differential equation in a hardware

graphics pipeline.

    b.   The art of Ewins is directed to MIP-Map selection for texture mapping (*Title*).

    c.   The art of Ewins and the art of Press as modified by Rumpf and Roy-Chowdhury are

analogous art because they both pertain to the art of computer graphics generation (*Ewins,*

Abstract; Rumpf, figure 3 and section 7, Anisotropic Diffusion in Image Processing).

    d.   Regarding **claim 7**:

    e.   Press does not specifically teach:

        i.   the local area of textures is generated by sampling a texture map;

    f.   Ewins appears to teach:

        i.   sampling a texture map (**pages 318 - 319, section 1.1 Texture Filtering**);

    g.   The motivation to use the art of Ewins with the art of Press as modified by Rumpf and Roy-

Chowdhury would have been the benefit recited in Ewins that texture mapping allows a high

degree of visual complexity without the expense of overly complex geometric modeling (**page**

317, section 1 Introduction, and Abstract), which would have been recognized as a benefit by the

ordinary artisan.

    h.   Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time

of invention to use the art of Ewins with the art of Press as modified by Rumpf and Roy-

Chowdhury to produce the claimed invention.

===

    i.   Regarding **claim 8**:

    j.   Press does not specifically teach:

        i.   the local area of textures is filtered;

k. Ewins appears to teach:

        i.   the local area of textures is filtered (**pages 318 - 319, section 1.1 Texture Filtering**);

==

l. Regarding **claim 9**:

m. Press does not specifically teach:

        i.   the local area of textures is filtered utilizing a plurality of filters;

n. Ewins appears to teach:

        i.   the local area of textures is filtered utilizing a plurality of filters (**pages 318 - 319, section 1.1 Texture Filtering**);

==

o. Regarding **claim 24**:

p. Press does not specifically teach:

        i.   the filtering is carried out using a programmable filter;

q. Ewins appears to teach:

        i.   the filtering is carried out using a programmable filter (**pages 318 - 319, section 1.1 Texture Filtering**);

==

r. Regarding **claim 25**:

s. Press does not specifically teach:

        i.   the filtering is carried out using a non-programmable filter;

t. Ewins appears to teach:

        i.   the filtering is carried out using a non-programmable filter (**pages 318 - 319, section 1.1 Texture Filtering**);

==

11. **Claims 10 - 11** are rejected under 35 U.S.C. 103(a) as being unpatentable over Press (Press, William H.; Flannery, Brian P.; Teukolsky, Saul A.; Vetterling, William T.; "Numerical Recipes in Fortran 77", 2001, Second edition, Cambridge University Press) in view of Rumpf (Martin Rumpf et al.; "Using Graphics Cards for Quantized FEM Computations", September 3 – 5 2001, Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing), further in view of Roy-Chowdhury (Roy-Chowdhury, Amber; Bellas, Nikolas; Banerjee, Prithviraj; "Algorithm-Based Error-Detection Schemes for Iterative Solution of Partial Differential Equations", 1996, IEEE Transactions on Computers, Vol. 45, No. 4).

    a.   Regarding **claim 10**:

    b.   Press appears to teach:

        i.   Receiving input (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that input is required to solve a partial differential equation, especially given the statement that an initial distribution relaxes to an equilibrium distribution on page 855**);

        ii.   Processing the input to generate the solution to the partial differential equation (**pages 854-856, section 19.5 Relaxation Methods for Boundary Value Problems**);

        iii.   Wherein the processing further includes determining whether the solution has converged (**pages 855, Relaxation Methods for Boundary Value Problems; second paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."**);

    c.   Press does not specifically teach:

        i.   Receiving input *in the hardware graphics pipeline*;

        ii.   Processing the input to generate the solution to the partial differential equation *utilizing the hardware graphics pipeline*;

        iii.   Generating output utilizing the hardware graphics pipeline for display;

        iv.   Wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline;

        v.   Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering

purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline;

vi.　　　*Wherein the input includes a local area of textures;*

vii.　　　*Wherein the local area of textures is filtered utilizing a filter including a plurality of elements;*

viii.　　*Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures;*

ix.　　　*Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors.*

d.　Rumpf appears to teach:

i.　　　Receiving input in the hardware graphics pipeline (*third page, figure 1*);

ii.　　　Processing the input to generate the solution to the partial differential equation utilizing the hardware graphics pipeline (*third page, section 3.1 Vector Representation, first paragraph; and seventh page, section 6. Linear Heat Equation, first paragraph*);

iii.　　Generating output utilizing the hardware graphics pipeline for display (*ninth page, figure 3*);

iv.　　Wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (*seventh page, section 6. Linear Heat Equation, first paragraph*);

v.　　　Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline (*ninth page, figure 3, displays surfaces and objects rendered by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline*).

vi.　　Wherein the input includes a local area of textures (*third page, figure 1; please note the textures input*);

vii.　　Wherein the local area of textures is filtered utilizing a filter including a plurality of elements (*seventh page, right-side column, second and third paragraphs; please note that a convolution operation is a filter operation*);

viii.   Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures (*second and third pages, section 2. Computational Setting; and third page, figure 1*).

e.   Roy-Chowdhury appears to teach:

i.   Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors (*page 395, right-side column, sentence that starts with, "The termination condition is determined . . .", and page 400, left-side column, top-half; please note that Press also appears to teach this limitation on page 855 second paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."*).

f.   The motivation to use the art of Rumpf with the art of Press would have been the benefits recited in Rumpf that the presented strategy opens a wide area of numerical applications for hardware acceleration (*first page, Abstract, first paragraph*), and turns a graphics card into an ultrafast vector coprocessor (*first page, Abstract, first paragraph*), which would have been recognized by the ordinary artisan as benefits that allow faster processing.

g.   The motivation to use the art of Roy-Chowdhury with the art of Press would have been the benefit recited in Roy-Chowdhury that the presented algorithm-based fault tolerance is an inexpensive method of achieving fault tolerance without requiring any hardware modifications, especially for iterative solution of linear systems arising from discretization of partial differential equations (**page 394, Abstract**), which would have been recognized as a benefit by the ordinary artisan.

h.   Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Rumpf with the art of Press to produce the claimed invention.

===

i.   Regarding claim 11:

i.   Claim 11 is taught as a subset of limitations as described in claim 10 above.

===

12. **Claims 26, 28 and 30 - 31** are rejected under 35 U.S.C. 103(a) as being unpatentable over Press (Press, William H.; Flannery, Brian P.; Teukolsky, Saul A.; Vetterling, William T.; "Numerical Recipes in C", 1988, Cambridge University Press) in view of Rumpf (Martin Rumpf et al.; "Using Graphics Cards for Quantized FEM Computations", September 3 - 5 2001, Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing), further in view of Roy-Chowdhury (Roy-Chowdhury, Amber; Bellas, Nikolas; Banerjee, Prithviraj; "Algorithm-Based Error-Detection Schemes for Iterative Solution of Partial Differential Equations", 1996, IEEE Transactions on Computers, Vol. 45, No. 4).

    a. Regarding **claim 26**:

    b. Press appears to teach:

        i.   Processing input (<u>pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that input is required to solve a partial differential equation, especially given the statement that an initial distribution relaxes to an equilibrium distribution on page 673</u>);

        ii.   Processing input to generate a solution to partial differential equations (<u>pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems</u>);

        iii.   Wherein the processing further includes determining whether the solution has converged (<u>page 674, Relaxation Methods for Boundary Value Problems; first paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."</u>);

    c. Press does not specifically teach:

        i.   *A hardware graphics pipeline for* processing input to generate a solution to partial differential equations *wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline*;

        ii.   *Wherein the graphics processing operation performed by the hardware graphics pipeline is enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline*;

      iii.   Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures;

      iv.   *Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors.*

d. Rumpf appears to teach:

      i.   A hardware graphics pipeline for processing input to generate a solution to partial differential equations wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (*third page, section 3.1 Vector Representation, first paragraph; and seventh page, section 6. Linear Heat Equation, first paragraph*);

      ii.   Wherein the graphics processing operation performed by the hardware graphics pipeline is enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline (*ninth page, figure 3, displays surfaces and objects rendered by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline*);

      iii.   Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures (*second and third pages, section 2. Computational Setting; and third page, figure 1*).

e. Roy-Chowdhury appears to teach:

      i.   Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors (*page 395, right-side column, sentence that starts with, "The termination condition is determined . . .", and page 400, left-side column, top-half; please note that Press also appears to teach this limitation on page 855 second paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."*).

f. The motivation to use the art of Rumpf with the art of Press would have been the benefits recited in Rumpf that the presented strategy opens a wide area of numerical applications for

hardware acceleration (*first page, Abstract, first paragraph*), and turns a graphics card into an

ultrafast vector coprocessor (*first page, Abstract, first paragraph*), which would have been

recognized by the ordinary artisan as benefits that allow faster processing.

g.   The motivation to use the art of Roy-Chowdhury with the art of Press would have been the

benefit recited in Roy-Chowdhury that the presented algorithm-based fault tolerance is an

inexpensive method of achieving fault tolerance without requiring any hardware modifications,

especially for iterative solution of linear systems arising from discretization of partial differential

equations (**page 394, Abstract**), which would have been recognized as a benefit by the ordinary

artisan.

h.   Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time

of invention to use the art of Rumpf and the art of Roy-Chowdhury with the art of Press to

produce the claimed invention.

==

i.   Regarding claim 28:

j.   Press appears to teach:

  i. Receiving boundary conditions (**pages 673-676, section 17.5 Relaxation Methods

  for Boundary Value Problems; it would have been obvious that boundary conditions

  are required to solve a partial differential equation, especially since the title of the

  section recites Boundary Value problems**);

  ii. Computing the solution to generate the solution to the partial differential

  equations involving the boundary conditions (**pages 673-676, section 17.5 Relaxation

  Methods for Boundary Value Problems**);

  iii. Determining whether the solution has converged (**page 674, first paragraph,

  subsection that starts with "Thus the algorithm . . .", sentence, "This procedure is then

  iterated until convergence."**);

  iv. If the solution has not converged, repeating the computing and determining

  (**page 674, first paragraph, subsection that starts with "Thus the algorithm . . .",

  sentence, "This procedure is then iterated until convergence."**);

  v. wherein the solution to the partial differential equation is generated (**pages 673-

  676, section 17.5 Relaxation Methods for Boundary Value Problems**);

k. Press does not specifically teach:

i.   Computing the solution to the partial differential equations involving the boundary conditions *at least some of the computing done in the hardware graphics pipeline*;

ii.   *Generating output utilizing the hardware graphics pipeline for display*;

iii.   wherein the solution to the partial differential equation is generated *utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline*;

iv.   *Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline*;

v.   *Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures* ;

vi.   *Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors*.

l. Rumpf appears to teach:

i.   Receiving input in the hardware graphics pipeline (*third page, figure 1*);

ii.   Computing the solution to the partial differential equations at least some of the computing done in the hardware graphics pipeline wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (*third page, section 3.1 Vector Representation, first paragraph; and seventh page, section 6. Linear Heat Equation, first paragraph*);

iii.   Generating output utilizing the hardware graphics pipeline for display (*ninth page, figure 3*);

iv.   Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline (*ninth page, figure 3, displays surfaces and objects rendered*

by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline);

v.   Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures (*second and third pages, section 2. Computational Setting; and third page, figure 1*).

m. Roy-Chowdhury appears to teach:

i.   Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors (*page 395, right-side column, sentence that starts with, "The termination condition is determined . . .", and page 400, left-side column, top-half; please note that Press also appears to teach this limitation on page 855 second paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."*).

n. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Rumpf and the art of Roy-Chowdhury with the art of Press to produce the claimed invention.

===

o. Regarding claim 30:

p. Press appears to teach:

i.   Receiving a first input (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that input is required to solve a partial differential equation, especially given the statement that an initial distribution relaxes to an equilibrium distribution on page 673**);

ii.   Processing the first input to generate a solution to a partial differential equation (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems**);

iii.   wherein the solution to the partial differential equation is generated (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems**);

iv.   Wherein the processing further includes determining whether the solution has converged (**page 674, Relaxation Methods for Boundary Value Problems; first**

paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This
procedure is then iterated until convergence.");

q. Press does not specifically teach:

   i.   Receiving a first input *into a hardware graphics pipeline*;

   ii.   Processing the first input to generate a solution to a partial differential equation
utilizing the hardware graphics pipeline;

   iii.   Receiving a second input into the hardware graphics pipeline;

   iv.   Rendering the 3D graphics image utilizing the hardware graphics pipeline for
display, wherein the rendering utilizes the second input and the result of the processing
of the first input;

   v.   wherein the solution to the partial differential equation is *generated utilizing the
hardware graphics pipeline for enhancing graphics processing operations performed by
the hardware graphics pipeline*;

   vi.   *Wherein the graphics processing operations performed by the hardware graphics
pipeline are enhanced by determining a location of surfaces or objects for rendering
purposes utilizing the solution to the partial differential equation generated utilizing the
hardware graphics pipeline*;

   vii.   *Wherein the input includes a local area of textures used to sample a texture map
to generate a modified local area of textures*;

   viii.   *Wherein the determining whether the solution has converged includes
calculating errors and concluding that the solution has converged based on the
calculation of the errors*.

r. Rumpf appears to teach:

   i.   Receiving a first input into a hardware graphics pipeline (*third page, figure 1*);

   ii.   Processing the first input to generate a solution to a partial differential equation
utilizing the hardware graphics pipeline (*eighth page, left-side column, second
paragraph; please note that an initial noisy image is input*);

   iii.   Receiving a second input into the hardware graphics pipeline (*eighth page, left-
side column, second paragraph; please note that a contrast enhancing function is input*);

iv.   Rendering the 3D graphics image utilizing the hardware graphics pipeline for display, wherein the rendering utilizes the second input and the result of the processing of the first input (*ninth page, figure 3*);

v.   wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (*seventh, eighth and ninth pages, section 7 Anisotropic Diffusion in Image Processing*);

vi.   Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline (*ninth page, figure 3, displays surfaces and objects rendered by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline*);

vii.   Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures (*second and third pages, section 2. Computational Setting; and third page, figure 1*).

s.   Roy-Chowdhury appears to teach:

i.   Wherein the determining whether the solution has converged includes calculating errors and concluding that the solution has converged based on the calculation of the errors (*page 395, right-side column, sentence that starts with, "The termination condition is determined . . .", and page 400, left-side column, top-half; please note that Press also appears to teach this limitation on page 855 second paragraph, section that starts with "Thus the algorithm consists . . .", sentence, "This procedure is then iterated until convergence."*).

t.   Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Rumpf and the art of Roy-Chowdhury with the art of Press to produce the claimed invention.

===

u. Regarding **claim 31**:

v. Press appears to teach:

    i.   The first input comprises boundary conditions (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that boundary conditions are required to solve a partial differential equation, especially since the title of the section recites Boundary Value problems**);

    ii.   The determining whether the solution has converged (**page 674, first paragraph, subsection that starts with "Thus the algorithm . . .", sentence, "This procedure is then iterated until convergence."**);

    iii.   If the solution has not converged, repeating the computing and determining (**page 674, first paragraph, subsection that starts with "Thus the algorithm . . .", sentence, "This procedure is then iterated until convergence."**);

===

13. **Claim 29** is rejected under 35 U.S.C. 103(a) as being unpatentable over Press (Press, William H.; Flannery, Brian P.; Teukolsky, Saul A.; Vetterling, William T.; "Numerical Recipes in C", 1988, Cambridge University Press), further in view of Roy-Chowdhury (Roy-Chowdhury, Amber; Bellas, Nikolas; Banerjee, Prithviraj; "Algorithm-Based Error-Detection Schemes for Iterative Solution of Partial Differential Equations", 1996, IEEE Transactions on Computers, Vol. 45, No. 4) further in view of Rumpf (Martin Rumpf et al.; "Using Graphics Cards for Quantized FEM Computations", September 3 – 5 2001, Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing).

a. Regarding **claim 29**:

b. Press appears to teach:

    i.   Receiving boundary conditions (**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems; it would have been obvious that boundary conditions are required to solve a partial differential equation, especially since the title of the section recites Boundary Value problems**);

ii.   computing the solution to the partial differential equation utilizing a relaxation

operation involving the boundary conditions (**pages 673-676, section 17.5 Relaxation**

Methods for Boundary Value Problems);

iii.   determining whether the solution has converged (**page 674, first paragraph,**

subsection that starts with "Thus the algorithm . . .", sentence, "This procedure is then

iterated until convergence.");

iv.   If the solution has not converged, repeating the computing and determining

(**page 674, first paragraph, subsection that starts with "Thus the algorithm . . .",**

sentence, "This procedure is then iterated until convergence.");

v.   if the solution has converged, incrementing a time value (**page 658, second**

paragraph, sentence that starts, "To solve equation (17.2.8) . . ."); and

vi.   repeating the foregoing operations using the incremented time value (**page 658,**

second paragraph, sentence that starts, "To solve equation (17.2.8) . . .").

vii.   wherein the solution to the partial differential equation is generated conditions

(**pages 673-676, section 17.5 Relaxation Methods for Boundary Value Problems**);

c. Press does not specifically teach:

i.   Receiving boundary conditions *in the form of at least one of geometry and*

textures;

ii.   computing the solution to the partial differential equation utilizing a relaxation

operation involving the boundary conditions *at least some of the computing done in the*

hardware graphics pipeline;

iii.   determining whether the solution has converged by:

(1) calculating the errors, ·

(2) summing the errors, and

iv.   concluding that the solution has converged if the sum of errors is less than a

predetermined amount;

v.   *Generating output utilizing the hardware graphics pipeline for display;*

vi.   wherein the solution to the partial differential equation is generated *utilizing the*

hardware graphics pipeline for enhancing graphics processing operations performed by

the hardware graphics pipeline;

vii.   *Wherein the graphics processing operations performed by the hardware graphics*

pipeline are enhanced by determining a location of surfaces or objects for rendering

purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline;

viii.   *Wherein the input includes a local area of textures used to sample a texture map to generate a modified local area of textures.*

d.   Roy-Chowdhury appears to teach:

    i.   determining whether the solution has converged by:

        (1) calculating the errors (**page 395, right-side column, sentence that starts with, "The termination condition is determined . . .", and page 400, left-side column, top-half**),

        (2) summing the errors (**page 395, right-side column, sentence that starts with, "The termination condition is determined . . .", and page 400, left-side column, top-half**), and

    ii.   concluding that the solution has converged if the sum of errors is less than a predetermined amount (**page 395, right-side column, sentence that starts with, "The termination condition is determined . . .", and page 400, left-side column, top-half**);

e.   Rumpf appears to teach:

    i.   Receiving boundary conditions in the form of at least one of geometry and textures (*third page, figure 1*);

    ii.   wherein the solution to the partial differential equation is generated utilizing the hardware graphics pipeline for enhancing graphics processing operations performed by the hardware graphics pipeline (*seventh page, section 6. Linear Heat Equation, first paragraph*);

    iii.   Wherein the graphics processing operations performed by the hardware graphics pipeline are enhanced by determining a location of surfaces or objects for rendering purposes utilizing the solution to the partial differential equation generated utilizing the hardware graphics pipeline (*ninth page, figure 3, displays surfaces and objects rendered by utilizing the solution to a partial differential equation utilizing a hardware graphics pipeline*).

    iv.   Generating output utilizing the hardware graphics pipeline for display (*ninth page, figure 3*);

> v.   Wherein the input includes a local area of textures used to sample a texture map
> to generate a modified local area of textures (*second and third pages, section 2.*
> *Computational Setting; and third page, figure 1*).

f.   The motivation to use the art of Roy-Chowdhury with the art of Press would have been the
benefit recited in Roy-Chowdhury that the presented algorithm-based fault tolerance is an
inexpensive method of achieving fault tolerance without requiring any hardware modifications,
especially for iterative solution of linear systems arising from discretization of partial differential
equations (<u>page 394, Abstract</u>).

g.   The motivation to use the art of Rumpf with the art of Press would have been the benefits
recited in Rumpf that the presented strategy opens a wide area of numerical applications for
hardware acceleration (*first page, Abstract, first paragraph*), and turns a graphics card into an
ultrafast vector coprocessor (*first page, Abstract, first paragraph*), which would have been
recognized by the ordinary artisan as benefits that allow faster processing.

h.   Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time
of invention to use the art of Roy-Chowdhury and the art of Rumpf with the art of Press to
produce the claimed invention.

==

14. **Examiner's Note:** Examiner has cited particular columns and line numbers in the references applied
to the claims above for the convenience of the applicant. Although the specified citations are
representative of the teachings of the art and are applied to specific limitations within the individual
claim, other passages and figures may apply as well. It is respectfully requested from the Applicant in
preparing responses, to fully consider the references in their entirety as potentially teaching all or part of
the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the
Examiner. <u>The entire body of all references are considered as being recited to teach the claimed</u>
<u>invention</u>.

Conclusion

15. The prior art made of record and not relied upon is relevant to the Applicant's disclosure:

    a.   Pan (U.S. Patent Number 6,078,938) teaches solving linear equations and repeating iterations a predetermined number of times.

    b.   E. Galligani et al.; "Implementation of Splitting Methods for Solving Block Tridiagonal Linear Systems on Transputers", 1995, Proceedings of Euromicro Workshop on Parallel and Distributed Processing, pages 409 – 415, especially page 412, left-side column, sentence that starts with, "The overheads can be minimized . . ."; teaches performing a convergence test after a predetermined number of iterations.

    c.   Olav Beckmann et al.; "Data Distribution at Run-Time: Re-Using Execution Plans", 1998, Euro-Par'98, LNCS 1470, Springer-Verlag, pages 413 – 421, especially page 418, text for Table 1, convergence test every 10 iterations; teaches performing a convergence test after a predetermined number of iterations.

    d.   Yulun Wang et al.; "A Processor Architecture for 3D Graphics", September 1992, IEEE Computer Graphics & Applications, pages 96 – 105, especially page 97, left-side column, partial differential equations; teaches using a graphics pipeline to solve partial differential equations.

    e.   Andrew S. Tanenbaum; "Structured Computer Organization", second edition, 1984, Prentice-Hall, pages 10 – 12; teaches hardware and software are equivalent.

    f.   Y. Saad; "Krylov Subspace Methods for Solving Large Unsymmetric Linear Systems", July 1981, Mathematics of Computation, Volume 37, Number 155, pages 105 - 126; teaches testing for convergence every q steps, page 113, and teaches calculating an error to determine when to stop.
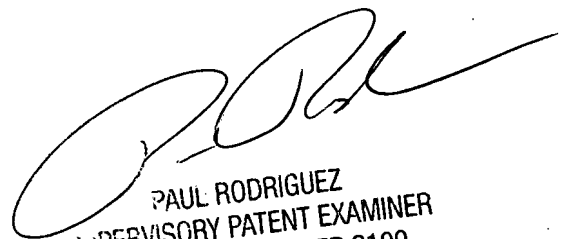
16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Russ Guill whose telephone number is 571-272-7955. The examiner can normally be reached on Monday – Friday 10:00 AM – 6:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached on 571-272-3753. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature or relating to the status of this application should be directed to the TC2100 Group Receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Russ Guill
Examiner
Art Unit 2123

RG

PAUL RODRIGUEZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100